

UDC 004.934

YU.YA. TOMKA, M.V. TALAKH, V.V. DVORZHAK, O.G. USHENKO

PRACTICAL ASPECTS OF FORMING TRAINING/TEST SAMPLES FOR CONVOLUTIONAL NEURAL NETWORKS

*Yuriy Fedkovych Chernivtsi National University,
2 Kotsjubynskiy Str. Chernivtsi, Ukraine, e-mail: y.tomka@chnu.edu.ua*

Анотація. Проаналізовані найбільш поширені підходи до оцінки якості навчання нейронних мереж у розрізі проблеми «малих навчальних вибірок». Проведений огляд кодових імплементацій найбільш універсальних підходів та способів розширення навчальних/тестових вибірок. Розглянуто логіку роботи STN-модуля, що може бути інкапсульований в існуючі згорткові нейронні мережі (CNN), надаючи можливість активного перетворення мапи ознак без додаткового навчання або модифікації процесу навчання.

Ключові слова: комп'ютерний зір, згорткова нейронна мережа, CNN, глибоке навчання, класифікація зображень, розуміння зображень

Abstract. The most common approaches to assessing the quality of training neural networks in the context of the problem of "small training sets" are analyzed. A review of the code implementation of the most universal approaches and ways of extending training/testing samples is carried out. The logic of the work of STN-module is analyzed. It can be inserted into existing convolutional architectures, giving neural networks the ability to actively spatially transform feature maps, conditional on the feature map itself, without any extra training supervision or modification to the optimization process.

Key Words: Computer Vision, Convolutional Neural Network, CNN, Deep Learning, Image Classification, Image Understanding

DOI: 10.31649/1681-7893-2022-43-1-24-35

INTRODUCTION

In recent years, convolutional neural networks (CNN) have achieved significant success in many areas of machine learning and computer vision [1, 2]. In addition, a wide range of various techniques for increasing the speed and quality of learning of convolutional neural networks has been developed [3, 4].

Researchers engaged in neural network modeling are often faced with the problem of a limited set of training data (for the field of computer vision - datasets of 2D images). This may be related both to the problems of finding the sources of this data and to the large dimension of the input and output variables of the problem being solved.

On the other hand, no less fundamental are the processes/algorithms of the "correct" formation of sets of educational data. Errors at this step are critical and ultimately can significantly reduce the effectiveness of neural network learning algorithms.

1. WAYS OF FORMING A SET OF TRAINING DATA

In the current literature on machine learning, as a rule, the issue of training/testing dataset formation is not always given sufficient attention or is completely ignored. Therefore, within the framework of our research, we should initially summarize the main and most common approaches and methods for evaluating the quality of the work of convolutional neural networks of arbitrary architecture precisely in the context of a "small training sample of data", where this problem is felt more acutely. Also, consider basic approaches to expanding training/test datasets.

1.1. WAYS TO ASSESS THE QUALITY OF LEARNING NEURAL NETWORKS ON SMALL DATA SAMPLES

Today, in general, in the tasks of training neural networks, depending on the task, different methods of assessing the quality of training are used:

1) cross-variation method [5]: one element is excluded from the sample one by one, without which the neural network is trained anew on the remaining elements. After training, a single error value is calculated for the excluded image. A similar operation is carried out with all other elements: they are excluded from the sample with a parallel return to it of previously excluded ones. Thus, each element of the dataset is considered both training and test, and knowing the errors of all test elements, they can be averaged.

This method provides a more accurate assessment of neural network testing than the direct use of a sample for testing that was previously used for training. The disadvantages of this approach are the need to repeat the training procedure multiple times and the possible inaccuracy of estimates of individual errors of the test elements due to the influence of the stochastic component of the training process.

[5] also provides a method for assessing the quality of neural network performance for the case of a small sample, which is called bootstrap. This method involves the multiple formation of different training samples based on a basic set of a small volume. At the same time, the data of the base sample may be used a different number of times, and some may not be used at all. For each of the samples formed in this way, an independent process of training the neural network and evaluating its error is carried out, after which the obtained data are generalized.

In [6], it is proposed to use the theory and methods of fuzzy logic and fuzzy sets to solve the problem of a "small amount of training data" in diffusion neural networks. These networks have a large number of input parameters, and, as a result, a very saturated structure. The approach involves drawing up vague "what if" rules to describe the relationships between various parameters, which allows for overcoming the problem of inconsistency and lack of data. With the use of these rules and methods of defuzzification, new data are obtained, due to which the existing training sample is expanded.

Another approach to solving the problem of a small amount of training data is proposed in [7], which consists in dividing the input and output variables into small groups, with the subsequent application to these groups of structurally much simpler artificial neural networks - single-layer perceptrons, for the training of which turns out to be enough of the available amount of data, and the subsequent unification of these perceptrons into a single structure of the perceptron complex.

1.2. METHODS OF EXPANDING EDUCATIONAL SAMPLES

The classic and simplest tool for improving the quality of neural network learning is dataset expansion. At the same time, it should be taken into account that the addition of data to the training sample should be carried out taking into account the fact that the more accurately the training sample approximates the general set of data that will be input to the neural network, the higher the quality of the training result will be.

The main methods of expanding initial selections:

1) Software generation. For the case of using synthetic training data, it is most convenient to generate missing training examples.

2) Data augmentation [8] – modification of existing images in order to expand training samples. As a rule, it comes down to compression/stretching, horizontal display, rotation, shift, and change of some pixels. The disadvantage of the method is the preservation of background regularities.

3) Hard sample mining [9]. A classic problem in image object search tasks is the need to maintain a sufficient number of training examples that look like the required object but are not.

4) Simulation of adding data. When training deep neural networks, the use of the dropout method [10] is considered a "good way": random zeroing of the activations of some neurons. That is, within the learning algorithm, the variability of the data is simulated - a modified version of the real image is randomly assigned to the input of the deeper levels of the network. Non-deficiency - the addition of such data that does not exist in reality can be simulated, due to which the accuracy of recognition may decrease.

5) Crowd sourcing. Since the training of deep neural networks requires significant volumes of manually labeled training samples, crowd sourcing services are used to form such samples [11].

When forming test sets, as a rule, the same problems arise as when forming training samples.

СИСТЕМИ ТЕХНІЧНОГО ЗОРУ І ШТУЧНОГО ІНТЕЛЕКТУ З ОБРОБКОЮ ТА РОЗПІЗНАВАННЯМ ЗОБРАЖЕНЬ

2. PRACTICAL ASPECTS OF FORMING TRAINING/TEST SAMPLES OF NEURAL NETWORKS

In [12, 13], a study of the possibility of recognizing road signs in Germany was conducted based on the GTSRB dataset [14]. The classic task of training a neural network for the classification of road signs from GTSRB was considered, as well as possible ways to improve the quality of work.

In [15] it is proposed to improve the obtained results [12, 13] through a two-step approach:

- 1) form train/validation/test data samples with preliminary image preprocessing;
- 2) use a combination of STN (spatial transformer network) [16] and IDSIA (convolutional neural network for traffic sign classification) [12] neural network architectures and analyze the impact of image preprocessing on learning quality.

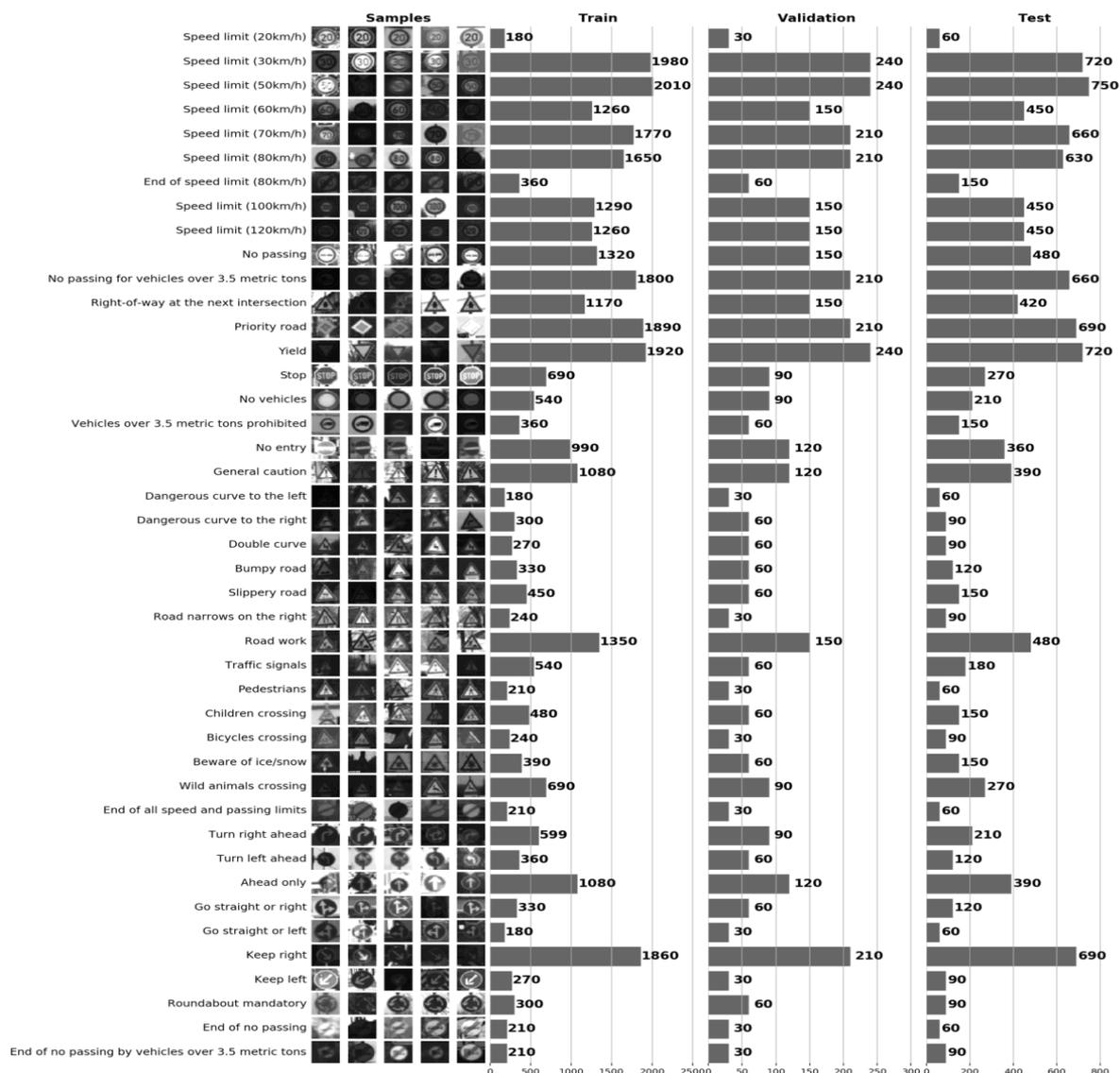


Figure 1 – Formed test/validation/test data samples

The analysis of the obtained results [12, 13, 15] showed that an easier way to get a generalized view of the data is to build histograms of the distributions of the train, validation and/or test datasets. At this stage, provided that development is conducted in the python language, using the matplotlib library is the most convenient tool. matplotlib.gridspec allows you to merge several graphs (in the study - 3 graphs at once) - and, as a result, get a complex graph that allows you to solve three current tasks "on the fly":

- Visualization of images. According to the graph, it is possible to evaluate a set of too dark/light images in the section of individual classes, based on which, image normalization can be carried out in order to eliminate the brightness variation.
- Checking the sample for imbalance. It is possible to estimate the prevalence of specimens of any class. To solve this problem, methods of undersampling or oversampling can be used [17].
- Checking the similarity of sample distributions (train, validation, and test, presented in Fig. 1). Spearman's rank correlation coefficients can also be used as a verification tool [18].

2.1. IMAGE NORMALIZATION

In order to improve the convergence of the neural network, it is advisable to carry out the procedure of bringing the images to a single color scale, as recommended in the article [13], in particular, in gray gradation.

This procedure is classically solved both with the help of OpenCV and with the help of the scikit-image python library, which can be easily installed using pip. OpenCV, in turn, requires independent compilation with a large number of dependencies, which is not very convenient.

Image contrast can be normalized using adaptive histogram equalization (CLAHE, contrast limited adaptive histogram equalization) [19].

It should be noted that skimage processes images in a single-threaded mode, which is obviously inefficient. In order to parallelize pre-processing of images, it is advisable to use the IPython Parallel library (ipyparallel). The approach to parallelization is simple: the sampling is divided into batches and each batch is processed independently of the others. As soon as all the batches are processed, there is a procedure to merge them back into one data set.

An example of images processed in this way is shown in Figure 2 and Figure 3.

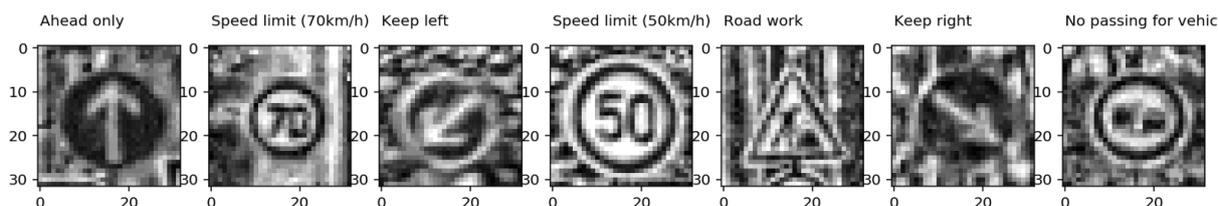


Figure 2 – The result of image normalization and conversion of their color gamut into grayscale

Choosing a larger interval in the CLASHE algorithm increases the contrast of the images, at the same time strongly highlighting their background, which makes the data noisy.

The values of this parameter can also be used to increase the contrast of the dataset in order to reduce the influence of background aspects on the retraining of the neural network itself.

2.2. DATA AUGMENTATION

It is absolutely clear that adding new and diverse data to the sample, as stated above, reduces the probability of retraining the neural network.

In general, it is possible to construct artificial images by transforming existing images using rotation, mirroring, and affine transformations.

Despite the fact that this process can be implemented for the entire sample, save the results, and then use them, however, a more elegant way is the procedure of creating new images "on the fly" (online) for the purpose of efficiency in adjusting the parameters of data augmentation. What, in particular, was demonstrated in the study [20].

Scaling and random rotations increase the size of the sample while preserving the belonging of the images to the ascending classes. Flips and rotations by 90, 180, 270 degrees can, on the contrary, change the meaning of the image (in the study, a road sign). In order to track such transitions, a table of possible transformations is offered for each road sign and the class they become as a result of such transformations. This table is classically represented as a (.csv file) [21].

In [15], the following functions of data augmentation in the problems of expanding training samples are proposed:

- customized affine transformations without rotation.
- random rotations and transformations to basic transformation tables (.csv file) that change data classes.

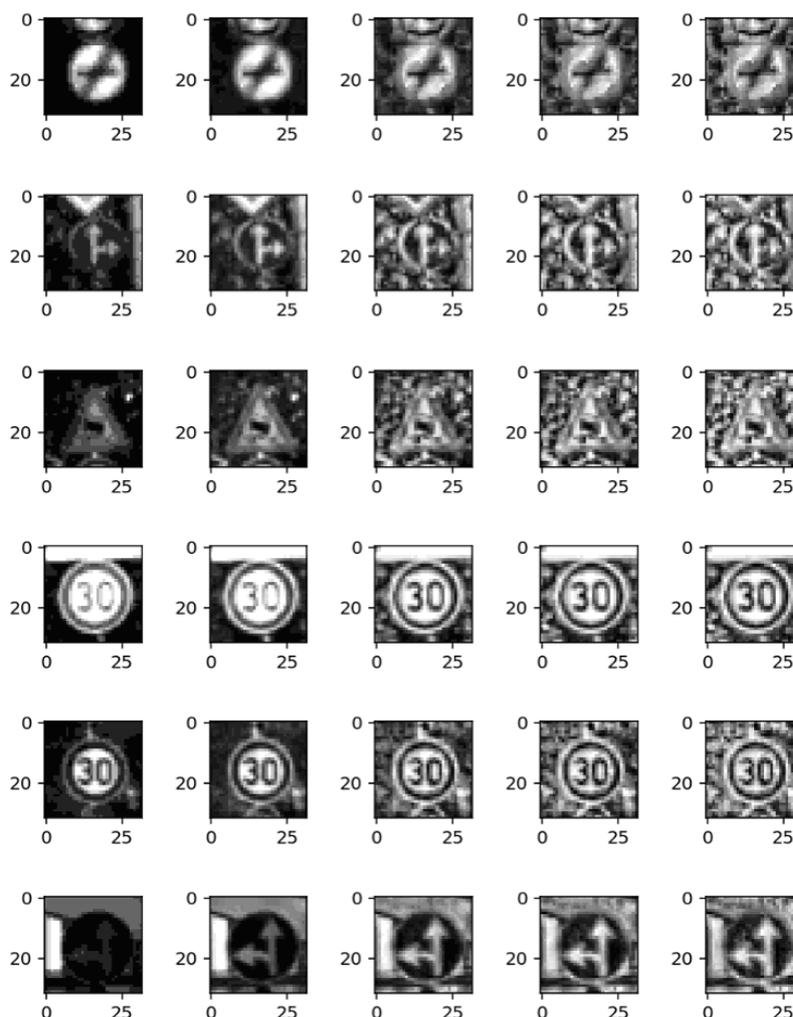


Figure 3 – Results of image normalization in gray scale for different intervals of the CLASHE algorithm: 8, 32, 128, 256, 512

The final step, the results of which are demonstrated in [15], is the combination of datasets with different parameters of the CLASHE algorithm into one large training sample and submission of the latter to the module of affine transformations. In other words, two types of transformations are implemented: contrast normalization and affine transformations, which are applied to the batch directly during the work process.

This approach allows you to preserve the distribution of classes on an extended set of test images compared to the ascending dataset.

3. IMAGE RECOGNITION USING CNN: A COMBINATION OF STN AND IDSIA ARCHITECTURES

To increase accuracy, an unpopular, but potentially effective approach to increasing accuracy should be considered. This technique consists of the use of a double convolutional neural network: STN (spatial transformer network), which receives processed image batches from the generator and focuses on road signs, and the IDSIA neural network recognizes the road sign in the received images from STN.

Thus, in [21], a possible software implementation of a neural network classifier of road signs with a possible option of encapsulation of the STN (Spatial Transformer Network) module is considered, on the basis of which the quality of the neural network is improved. STN, applying a learning affine transformation followed by interpolation, deprives the image of spatial invariance. In other words, the purpose of using STN is to rotate or reduce/enlarge the ascending image in such a way that the main classifier network can more easily identify the desired object (Fig. 4).

The STN block can be placed in a convolutional neural network (CNN), working in it autonomously, learning on the gradients coming from the main network.

3.1. GENERAL PRINCIPLE OF STN

One of the problems of convolutional neural networks is low invariance to input data: different scale, shooting point, background noise, etc. [23]. Of course, the pooling operation provides some invariance, but, in fact, it simply reduces the size of the feature map, resulting in a loss of information.

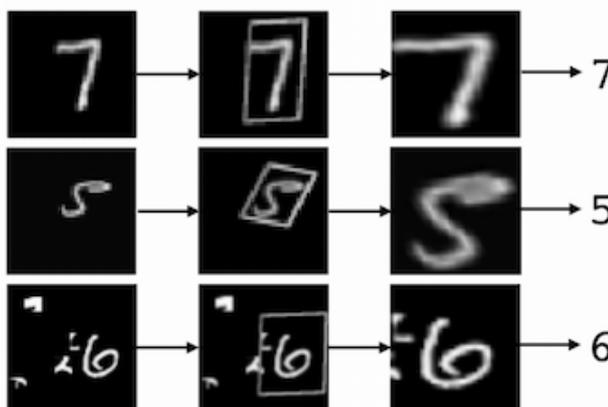


Fig. 4 – An example from the MNIST dataset from [25]. Cluttered MNIST (left), target object recognized by STN (center), transformed image (right)

Unfortunately, due to the small receptive field in standard 2x2 pooling, spatial invariance can be achieved only in deep layers close to the output layer. Also, pooling does not ensure rotation and scaling invariance [24].

The main and most common way to make the model resistant to these variations is the augmentation of the dataset, the main methods of implementation of which were discussed above with possible methods of automation [22].

3.2. STN MODULE OPERATION ALGORITHM

The operation of the STN module can be reduced to the process (not including the training stage) shown in Figure 5.

The generalized STN transformation algorithm is as follows [16]:

Step 1. Determine the affine transformation matrix Θ , which directly describes the transformation (each affine transformation has its own matrix). We are interested in the following four situations:

- identical transformation (the output is the same image);
- counter-clockwise rotation;
- approaching the center twice;
- twice as far from the center.

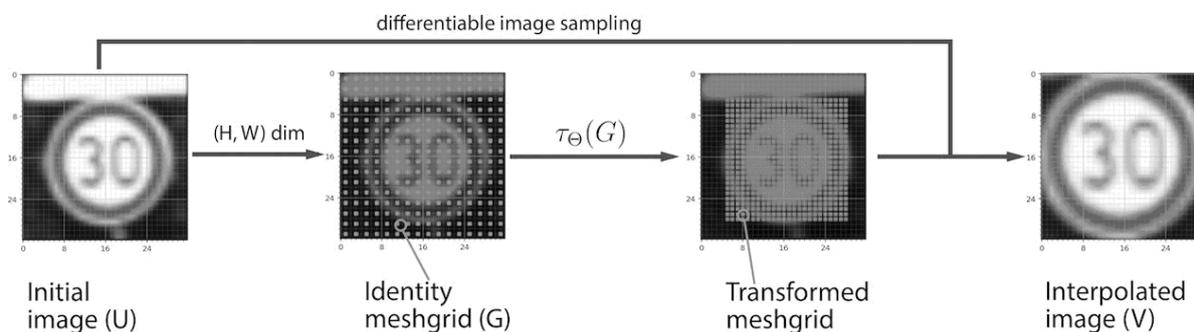


Figure 5 – Application of STN-transformation in 4 steps with a known matrix of linear transformations Θ

Step 2. Instead of applying the transformation directly to the upstream image (U), a sampling meshgrid (G) of the same size as U is created. The sampling grid is a set of indices (x_t, y_t) , which cover the ascending image space. The grid does not contain any information about the color of the images.

Step 3. Apply the matrix of linear transformations to the created sampling grid to obtain a new set of points on the grid, each of which can be determined as the result of multiplying the matrix Θ by the coordinate vector (x_t, y_t) , with a free term:

$$\begin{bmatrix} x_i^s \\ y_i^s \end{bmatrix} = \theta \begin{bmatrix} x_i^t \\ y_i^t \\ 1 \end{bmatrix} \quad (1)$$

Step 4. Obtain a subsample V using an ascending feature map, a transformed work grid (step 3) and a differentiated interpolation function of your choice (for example, bilinear). Interpolation is necessary, as it is necessary to translate the sampling result (potentially possible fractional pixel values) into whole numbers.

$$V_i^c = \sum_n^H \sum_m^W U_{nm}^c k(x_i^s - m; \Phi_x) k(y_i^s - n; \Phi_y) \quad \forall i \in [1 \dots H'W'] \quad \forall c \in [1 \dots C]$$

$\forall i \in [1 \dots H'W']$ pixel in a channel c $\forall c \in [1 \dots C]$ channels

x coordinate in $\tau_\theta(G)$ parameters of sampling kernel
 value at location (n, m) in channel c of input U interpolation kernel

Fig. 6 – Sampling and interpolation

Learning task. Generally speaking, if the values of Θ for each output image were known in advance, the process described above could be started. In fact, it would be appropriate to obtain Θ from the data with the help of machine learning. It is quite possible to do this.

First, it is necessary to make sure that the loss function of the traffic image classifier can be minimized using back propagation logic through the sampler.

Secondly, there are gradients along U and G : that is why the interpolation function must be differentiated or, at least, partially differentiated. Third, the partial derivatives of x and y with respect to Θ are calculated.

Finally, a LocNet (localizing regressor network) is created, the only task of which will be to learn and predict the correct Θ for the input image, using the loss function that was minimized through the common backprop (Figure 7).

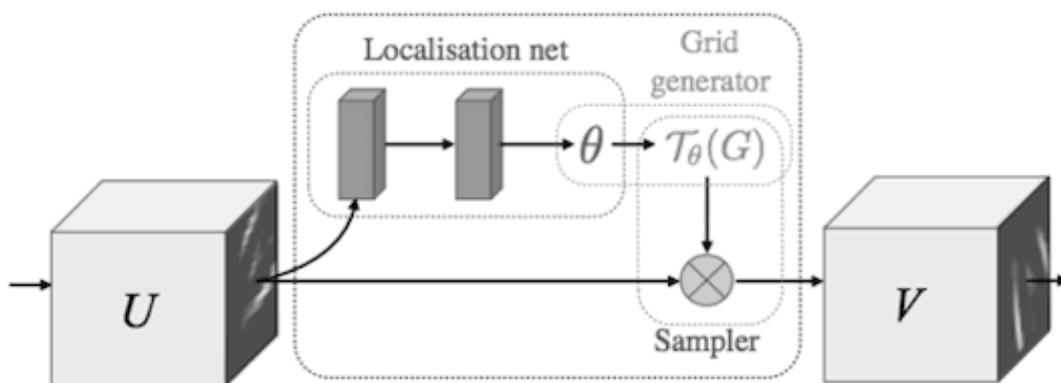


Figure 7 – STN scheme [25]

The main advantage of this approach is that a differentiated autonomous module with memory (in the form of learning weights) is obtained, which can be placed in any part of the CNN.

Thus, all stages of STN creation were analyzed: creation of LocNet, mesh grid generator and sampler.

It is appropriate to conduct a generalized analysis of construction and training on Tensor Flow in accordance with the considered algorithm.

4. BUILDING A MODEL IN TENSORFLOW

The ultimate goal is to teach how to recognize images (in the previously analyzed sources – road signs), and to achieve this, you need to create a classifier and train it. Therefore, at the first stage, it is necessary to choose a classifier network (it can be arbitrary). The version of the INDSIA network [26], implemented by Torch, was considered as an approbation of the methods under consideration.

At the second stage, it is necessary to define and teach the STN module, which, accepting an ascending image as an input, transforms it with the help of a sampler and a new image is obtained at the output (or a minibatch, if the work is carried out in batch mode), which in turn is used by the classifier. It should be noted that STN can be easily excluded from the calculation schedule by replacing the entire module with a simple batch generator. In this case, the usual classifier network is obtained.

The general scheme of the resulting double neural network is presented in Fig. 8.

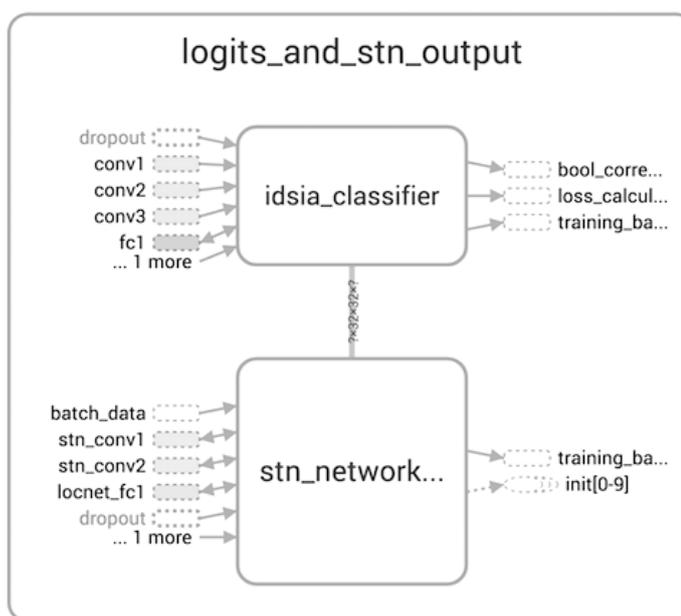


Figure 8 – STN transforms the input images and submits them to the input of IDSIA, which learns with the help of backprop and then performs their classification

The results of the activation function of each convolutional layer are combined into one vector, which is then transmitted to fully connected layers (Figure 9). This is an example of so-called multiscale features that additionally improve the quality of the classifier. The converted STN image is fed to the conv1 input.

Now that the log calculation method (STN + IDSIA network) is known, the next step is to optimize the loss function (as a function of which it is advisable to use cross-entropy or log loss).

The structure of the classifier itself is shown in Figure 9.

You should pay attention to the network initialization task with the value of the learning rate parameter. A correctly selected value allows you to achieve the fact that the gradients will be able to spread information faster to the LocNet STN, which is located in the outer layers of the entire neural network. In the opposite case, this network will learn more slowly due to the "sliding gradient" problem [27]. Also, it is necessary to remember that too small upward learning rate values do not allow the neural network to approximate image elements well.

The general problem of using an STN module with a CNN is the need to ensure that both networks are not retrained, which makes the training process itself complex and unstable. However, on the other hand, adding a small amount of augmented data (especially the argumentation of brightness) to the training sample allows you to significantly reduce the level of retraining. In any case, the advantages are more tangible than the disadvantages: even without augmentation, good results are observed, and STN + IDSIA surpasses the accuracy of IDSIA without this module by 0.5-1%.

СИСТЕМИ ТЕХНІЧНОГО ЗОРУ І ШТУЧНОГО ІНТЕЛЕКТУ З ОБРОБКОЮ ТА РОЗПІЗНАВАННЯМ ЗОБРАЖЕНЬ

Already after 10 training epochs, an accuracy of 99.12% is achieved on the validation data set. The CNN is retrained, but it should be noted that a double-complex mesh was used on the original dataset without augmentation. By adding augmentation, it is possible to obtain an accuracy of 99.71% on the validation set after 10 iterations, but with a significant increase in training time (Figure 10).

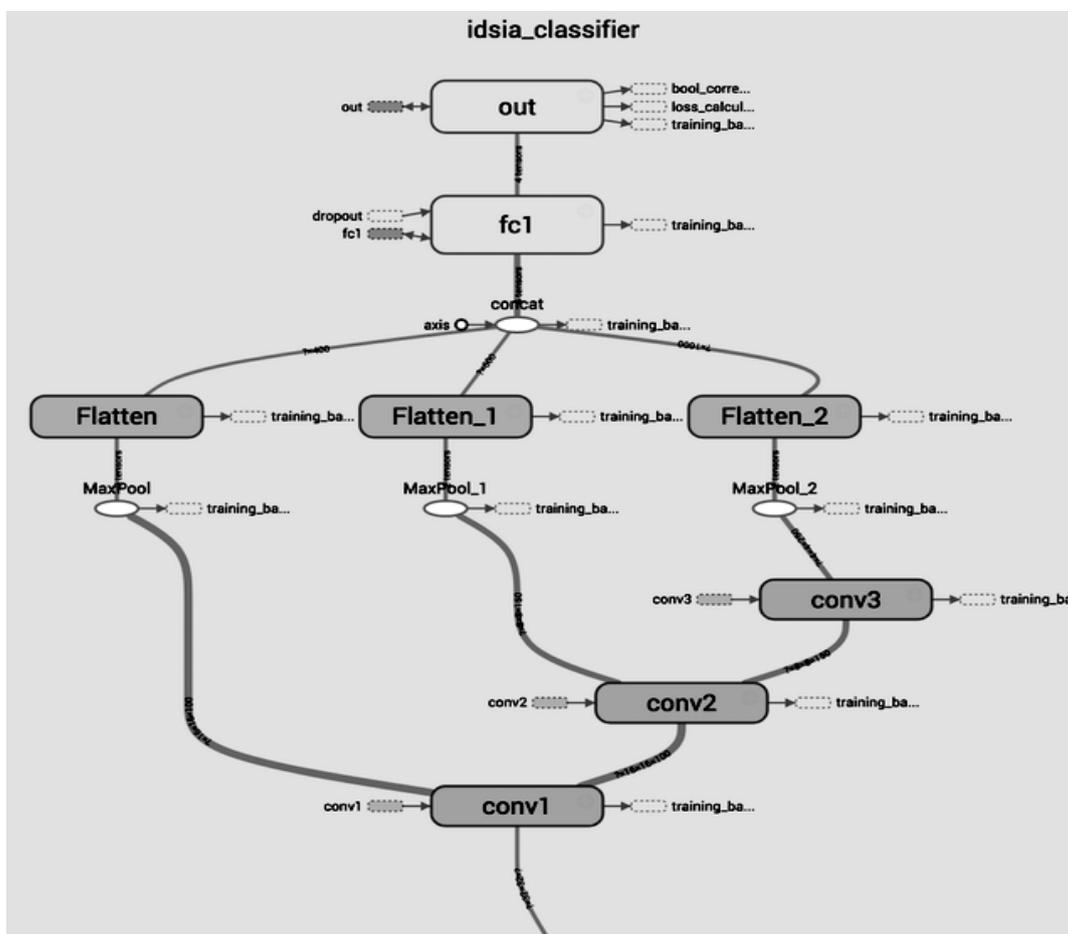


Figure 9 – Computational graph of the IDSIA network

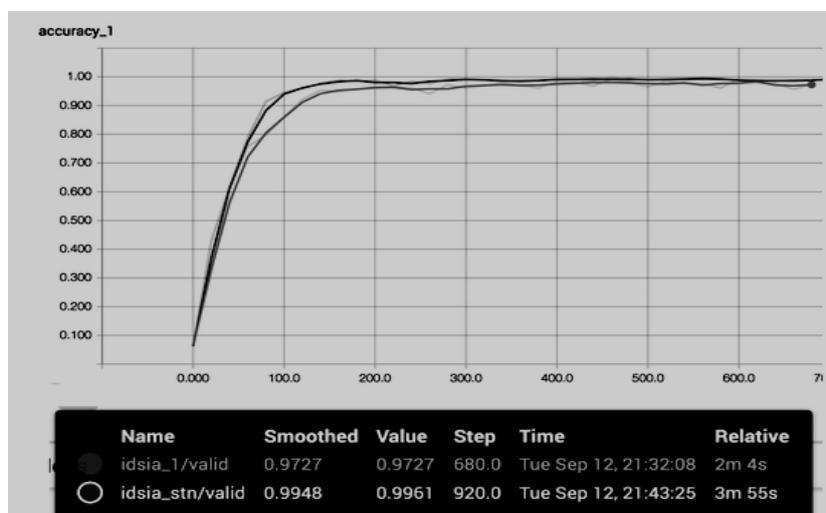


Figure 10 – Model training results (idsia_1 is IDSIA network without IET module, STN+IDSIA)

RESULTS

The most common approaches to assessing the quality of learning neural networks in terms of the problem of "small training samples" are analyzed.

An overview of the code implementations of the most universal approaches and methods of expanding (adding images) training/test samples based on the python language was conducted.

The potential of the approach, which consists of the use of a double convolutional neural network: STN (spatial transformer network), which receives processed image batches from the generator and focuses on image elements and the IDSIA neural network, which recognizes the image (road-th sign), which are received from STN.

The software-differentiated STN module can be considered as an alternative to image augmentation, which is a standard way to achieve spatial invariance for CNNs by applying affine transformations to input images (feature maps). It can be integrated into a convolutional neural network by placing it immediately after the batch generator, with the goal of learning the transformation matrix Θ , which minimizes the loss function of the main classifier (the main neural network in our case).

Adding STN to CNN complicates training, and makes it unstable: there is a problem of control so that both networks (instead of one) do NOT retrain. This fact, as of today, is the main aspect of why similar LEGO modules have not reached widespread distribution. This issue requires additional research and the appropriate search for ways to speed up the operation of the STN-CNN connection.

STNs trained on augmented data, especially for the case of luminance augmentation, demonstrate better quality and low retraining.

СПИСОК ЛІТЕРАТУРИ

1. *Krizhevsky A.*, Imagenet classification with deep convolutional neural networks [Electronic resource] / A.Krizhevsky, I. Sutskever, G. Hinton – 2012. – Access Mode: <https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>
2. *Szegedy C.* Going deeper with convolutions [Electronic resource] / C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich – 2015. – Access Mode: <https://arxiv.org/pdf/1409.4842.pdf>
3. *Zeiler M.D.* Stochastic pooling for regularization of deep convolutional neural networks [Electronic resource] / M. D. Zeiler, R. Fergus – 2015. – Access Mode: <https://arxiv.org/pdf/1301.3557.pdf>
4. *Mahendran A.* Understanding deep image representations by inverting them networks [Electronic resource] / A. Mahendran, A. Vedaldi – 2015. – Access Mode: <https://arxiv.org/pdf/1312.4400.pdf>
5. Huang C. A Diffusion-Neural-Network for Learning from Small Samples/ C. Huang, C. Moraga // International Journal of Approximate Reasoning. – 2004. – V. 35 P. 137–161.
6. *Wang J.* The Effectiveness of Data Augmentation in Image Classification using Deep Learning networks [Electronic resource] / Wang J., Perez L. Vedaldi – 2017. – Access Mode: <http://cs231n.stanford.edu/reports/2017/pdfs/300.pdf>
7. *Canavet O.* Efficient sample mining for object detection / O. Canavet, F. Fleuret // Proceedings of the Asian Conference on Machine Learning (ACML) – 2014 – P. 48-63.
8. *Srivastana N.* A Simple Way to Prevent Neural Networks From Overfitting / N. Srivastana, G. Hinton, A. Krizhevsky та ін. // The Journal of Machine Learning Research. – 2014. – Vol. 15, №1. – С. 1929–1958.
9. Amazon Mechanical Turk (MTurk) [Electronic resource] – Access Mode: <https://www.mturk.com/>.
10. *Cires,an D.* Multi-column Deep Neural Networks for Image Classification [Electronic resource] / D. Cires,an, U. Meier, J. Schmidhuber. – 2012. – Access Mode: <https://arxiv.org/pdf/1202.2745.pdf>.
11. *LeCun Y.* Traffic Sign Recognition with Multi-Scale Convolutional Networks [Electronic resource] / Y. LeCun, P. Sermanet – Access Mode: <http://yann.lecun.com/exdb/publis/pdf/sermanet-ijcnn-11.pdf>.
12. German Traffic Sign Recognition Benchmark [Electronic resource] – Access Mode: <http://benchmark.ini.rub.de/?section=gtsrb&subsection=dataset>
13. Розпізнавання дорожніх знаків з допомогою CNN: Інструменти для препроцесинга зображень [Electronic resource]. – 2017. – Access Mode: <https://habrahabr.ru/company/newprolab/blog/334618/>.
14. *Jaderberg M.* Spatial Transformer Networks [Electronic resource] / M.Jaderberg, K. Simonyan, A. Zisserman, K. Kavukcuoglu. – 2016. – Access Mode: <https://arxiv.org/pdf/1506.02025.pdf>.
15. Class Imbalance Problem [Electronic resource]. – 2013. – Access Mode: <http://www.chioka.in/class-imbalance-problem/>.
16. Spearman's rank correlation coefficient [Electronic resource] – Access Mode: https://en.wikipedia.org/wiki/Spearman%27s_rank_correlation_coefficient.

СИСТЕМИ ТЕХНІЧНОГО ЗОРУ І ШТУЧНОГО ІНТЕЛЕКТУ З ОБРОБКОЮ ТА РОЗПІЗНАВАННЯМ ЗОБРАЖЕНЬ

17. *Zuiderveld K.* Contrast Limited Adaptive Histogram Equalization / Karel Zuiderveld. // *Graphic Gems IV*. San Diego: Academic Press Professional. – 1994. – С. 474–485.
18. *Danilyuk K.* ConvNets Series. Image Processing: Tools of the Trade [Electronic resource] / Kiril Danilyuk. – 2017. – Access Mode: <https://towardsdatascience.com/convnets-series-image-processing-tools-of-the-trade-36e168836f0c>.
19. Распознавание дорожных знаков с помощью CNN: Spatial Transformer Networks [Electronic resource]. – 2017. – Access Mode: <https://habrahabr.ru/company/newprolab/blog/339484/>.
20. CS231n Convolutional Neural Networks for Visual Recognition [Electronic resource] – Access Mode: <http://cs231n.github.io/classification/>.
21. *Zakka K.* Deep Learning Paper Implementations: Spatial Transformer Networks [Electronic resource] / Kevin Zakka. – 2017. – Access Mode: <https://kevinzakka.github.io/2017/01/18/stn-part2/>.
22. The MNIST - Database of handwritten digits [Electronic resource] – Access Mode: <http://yann.lecun.com/exdb/mnist/>.
23. Traffic sign recognition with Torch [Electronic resource]. – 2015. – Access Mode: <https://github.com/Moodstocks/gtsrb.torch>.
24. Vanishing gradient problem [Electronic resource] – Access Mode: https://en.wikipedia.org/wiki/Vanishing_gradient_problem.
25. Olexander N. Romanyuk, Sergii V. Pavlov, and etc. "A function-based approach to real-time visualization using graphics processing units", Proc. SPIE 11581, Photonics Applications in Astronomy, Communications, Industry, and High Energy Physics Experiments 2020, 115810E (14 October 2020); <https://doi.org/10.1117/12.2580212>.
26. Leonid I. Timchenko, Natalia I. Kokriatskaia, Sergii V. Pavlov, and etc. "Q-processors for real-time image processing", Proc. SPIE 11581, Photonics Applications in Astronomy, Communications, Industry, and High Energy Physics Experiments 2020, 115810F (14 October 2020); <https://doi.org/10.1117/12.2580230>

REFERENCES

1. *Krizhevsky A.*, Imagenet classification with deep convolutional neural networks [Electronic resource] / A.Krizhevsky, I. Sutskever, G. Hinton – 2012. – Access Mode: <https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>
2. *Szegedy C.* Going deeper with convolutions [Electronic resource] / C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich – 2015. – Access Mode: <https://arxiv.org/pdf/1409.4842.pdf>
3. *Zeiler M.D.* Stochastic pooling for regularization of deep convolutional neural networks [Electronic resource] / M. D. Zeiler, R. Fergus – 2015. – Access Mode: <https://arxiv.org/pdf/1301.3557.pdf>
4. *Mahendran A.* Understanding deep image representations by inverting them networks [Electronic resource] / A. Mahendran, A. Vedaldi – 2015. – Access Mode: <https://arxiv.org/pdf/1312.4400.pdf>
5. Huang C. A Diffusion-Neural-Network for Learning from Small Samples/ C. Huang, C. Moraga // *International Journal of Approximate Reasoning*. – 2004. – V. 35 P. 137–161.
6. *Wang J.* The Effectiveness of Data Augmentation in Image Classification using Deep Learning networks [Electronic resource] / Wang J., Perez L. Vedaldi – 2017. – Access Mode: <http://cs231n.stanford.edu/reports/2017/pdfs/300.pdf>
7. *Canavet O.* Efficient sample mining for object detection / O. Canavet, F. Fleuret // *Proceedings of the Asian Conference on Machine Learning (ACML)* – 2014 – P. 48-63.
8. *Srivastana N.* A Simple Way to Prevent Neural Networks From Overfitting / N. Srivastana, G. Hinton, A. Krizhevsky та ін.]. // *The Journal of Machine Learning Research*. – 2014. – Vol. 15, №1. – С. 1929–1958.
9. Amazon Mechanical Turk (MTurk) [Electronic resource] – Access Mode: <https://www.mturk.com/>.
10. *Cires, an D.* Multi-column Deep Neural Networks for Image Classification [Electronic resource] / D. Cires, an, U. Meier, J. Schmidhuber. – 2012. – Access Mode: <https://arxiv.org/pdf/1202.2745.pdf>.
11. *LeCun Y.* Traffic Sign Recognition with Multi-Scale Convolutional Networks [Electronic resource] / Y. LeCun, P. Sermanet – Access Mode: <http://yann.lecun.com/exdb/publis/pdf/sermanet-ijcnn-11.pdf>.
12. German Traffic Sign Recognition Benchmark [Electronic resource] – Access Mode: <http://benchmark.ini.rub.de/?section=gtsrb&subsection=dataset>
13. Traffic Sign Recognition with CNN: Tools for Image Preprocessing [Electronic resource]. - 2017. - Access mode: <https://habrahabr.ru/company/newprolab/blog/334618/>.

СИСТЕМИ ТЕХНІЧНОГО ЗОРУ І ШТУЧНОГО ІНТЕЛЕКТУ З ОБРОБКОЮ ТА РОЗПІЗНАВАННЯМ ЗОБРАЖЕНЬ

14. *Jaderberg M.* Spatial Transformer Networks [Electronic resource] / M.Jaderberg, K. Simonyan, A. Zisserman, K. Kavukcuoglu. – 2016. – Access Mode: <https://arxiv.org/pdf/1506.02025.pdf>.
15. Class Imbalance Problem [Electronic resource]. – 2013. – Access Mode: <http://www.chioka.in/class-imbalance-problem/>.
16. Spearman's rank correlation coefficient [Electronic resource] – Access Mode: https://en.wikipedia.org/wiki/Spearman%27s_rank_correlation_coefficient.
17. *Zuiderveld K.* Contrast Limited Adaptive Histogram Equalization / Karel Zuiderveld. // Graphic Gems IV. San Diego: Academic Press Professional. – 1994. – С. 474–485.
18. *Danilyuk K.* ConvNets Series. Image Processing: Tools of the Trade [Electronic resource] / Kiril Danilyuk. – 2017. – Access Mode: <https://towardsdatascience.com/convnets-series-image-processing-tools-of-the-trade-36e168836f0c>.
19. Traffic sign recognition with CNN: Spatial Transformer Networks [Electronic resource]. - 2017. - Access Mode: <https://habrahabr.ru/company/newprolab/blog/339484/>.
20. CS231n Convolutional Neural Networks for Visual Recognition [Electronic resource] – Access Mode: <http://cs231n.github.io/classification/>.
21. *Zakka K.* Deep Learning Paper Implementations: Spatial Transformer Networks [Electronic resource] / Kevin Zakka. – 2017. – Access Mode: <https://kevinzakka.github.io/2017/01/18/stn-part2/>.
22. The MNIST - Database of handwritten digits [Electronic resource] – Access Mode: <http://yann.lecun.com/exdb/mnist/>.
23. Traffic sign recognition with Torch [Electronic resource]. – 2015. – Access Mode: <https://github.com/Moodstocks/gtsrb.torch>.
24. Vanishing gradient problem [Electronic resource] – Access Mode: https://en.wikipedia.org/wiki/Vanishing_gradient_problem.
25. Olexander N. Romanyuk, Sergii V. Pavlov, and etc. "A function-based approach to real-time visualization using graphics processing units", Proc. SPIE 11581, Photonics Applications in Astronomy, Communications, Industry, and High Energy Physics Experiments 2020, 115810E (14 October 2020); <https://doi.org/10.1117/12.2580212>.
26. Leonid I. Timchenko, Natalia I. Kokriatskaia, Sergii V. Pavlov, and etc. "Q-processors for real-time image processing", Proc. SPIE 11581, Photonics Applications in Astronomy, Communications, Industry, and High Energy Physics Experiments 2020, 115810F (14 October 2020); <https://doi.org/10.1117/12.2580230>
- 27.

Надійшла до редакції 15.02.2022р.

TOMKA YURIY – Ph.D., assistant professor of Computer Science Department, Yuriy Fedkovich Chernivtsi National University, Chernivtsi, Ukraine, [e-mail: y.tomka@chnu.edu.ua](mailto:y.tomka@chnu.edu.ua)

TALAKH MARIA – Ph.D., assistant professor of Computer Science Department, Yuriy Fedkovich Chernivtsi National University, Chernivtsi, Ukraine, [e-mail: m.talah@chnu.edu.ua](mailto:m.talah@chnu.edu.ua)

DVORZHAK VALENTINA – Ph.D., assistant professor of Computer Science Department, Yuriy Fedkovich Chernivtsi National University, Chernivtsi, Ukraine, [e-mail: v.dvorzhak@chnu.edu.ua](mailto:v.dvorzhak@chnu.edu.ua)

USHENKO OLEXANDER – D.Sc., Professor, Head of Optics and Publishing Department, Yuriy Fedkovych Chernivtsi National University, Chernivtsi, Ukraine, [e-mail: o.ushenko@chnu.edu.ua](mailto:o.ushenko@chnu.edu.ua)